ACEEE Int. J. on Electrical and Power Engineering , Vol. 4, No. 2, Aug 2013

Cluster Computing Environment for On - line Static Security Assessment of large Power Systems

Sunitha R[1], Sreerama Kumar R.[2], Abraham T. Mathew[1] and Veeresh P. Kosaraju[3]
[1]Electrical Engineering Department, NIT Calicut, Kerala, India. Email: {rsunitha, atm}@nitc.ac.in
[2]King Abulaziz University Jeddah, Saudi Arabia. Email: sreeram@nitc.ac.in
[3]Coal India limited, Maharashtra, India. Email: veeresh_kosaraju@yahoo.co.in

***Abstract*— The increased size of modern power systems demand faster and accurate means for the security assessment, so that the decisions for reliable and secure operation planning could be drawn in a systematic manner. Large computational overhead is the major impediment in preventing the power system security assessment (PSSA) from on-line use. To mitigate this problem, this paper proposes, a cluster computing based architecture for power system static security assessment, utilizing the tools in the open source domain. A variant of the master/slave pattern is used for deploying the cluster of workstations (COW), which act as the computational engine for the on-line PSSA. The security assessment is performed utilizing the developed composite security index that can accurately differentiate the secure and non-secure cases and has been defined as a function of bus voltage and line flow limit violations. Due to the inherent parallel structure of security assessment algorithm and to exploit the potential of distributed computing, domain decomposition is employed for parallelizing the sequential algorithm. Extensive experimentations were carried out on IEEE 57 bus and IEEE 145-bus 50 machine standard test systems for demonstrating the validity of the proposed architecture.**

***Index Terms*—first term, second term, third term, fourth term, fifth term, sixth term**

I. Introduction

Modern society critically relies on a securely operated electric power system for electricity. By nature, a power system is continually experiencing disturbances (contingencies), such as load changes, outage of generators or other equipment, short circuits, or combination of such events. These disturbances usually lead to changes in the configuration and/or state of the power system. Security refers to the degree of risk in a power system's ability to survive imminent disturbances without interruption to customer service at any instant of time [1]. With the initiation of the deregulated electricity market, the system operators are concerned with the special measures to protect the system against severe contingences and to increase the security margins. These actions are performed by them based on the results obtained by conducting power system security analysis The calculations required for the power system security assessment are performed based on the (n-1) criterion that requires the analysis of system behavior and the verification of operational limits violations for each credible contingency. Traditionally these analyses are carried out off -line as it requires the solution of system state equations in both static and dynamic time frame. These off-line analyses referred to as worst case scenarios, give operational limits often that are too restrictive or, in the case when the real time conditions differ to the reference values, highly conservative [2]. Therefore, these analyses appear to be inadequate in the new competitive scenario where there is an uncertainty in predicting the future operating conditions. This trend has increased the need for fast and more accurate methods of security assessment [1].

In the new competitive environment, the utilities are forced to conduct the real time power system security assessment, in which the security is assessed in real time for a large set of probable contingencies and transactions [3]. The real-time analysis could lead to a credible improvement of the utilization of the available infrastructure at adequate reliability levels allowing system operators to obtain more realistic operational guidance in planning preventive and corrective actions aimed to mitigate the effect of critical contingencies [1-2]. Traditional sequential computation is inadequate for on-line power system security analysis as the entire computation should take, typically less than a few minutes for the information to be useful [4]. The application of artificial intelligent [5] and probabilistic [6] based methodologies have been attempted for obtaining fast but less accurate solution for security assessment.

Considerable research efforts [7]-[9] have also been oriented to develop dedicated computer architectures based on supercomputers or network of workstations for the fast solution of power system state equations. This method is applied particularly to on-line power system security assessment, where it is necessary to predict the impact of credible contingencies and suggest suitable preventive or corrective control actions within a few minutes to mitigate the effects of critical events. In recent years parallel processing based on distributed systems seems to be a viable solution to speed up the simulations in order to obtain results in useful time. Security constrained optimal power flow solution in a distributed computing environment is proposed in [8]. In [9] the various functions of security analysis are mapped on to a network of workstations which work as a continuous flow of base case conditions. As supporting tools in developing this activity, the application of TCP/IP based communication services and web based control architectures have been recently published in [2].

This work proposed in this paper mainly focuses on power system static security assessment, contingency screening and ranking. Contingency screening and ranking is conventionally performed by computing a scalar performance index (PI), derived from DC or fast decoupled load flow solution for

DOI: 01.IJEPE.4.2.1283

each contingency [10]. These methods generally employ a quadratic function as the performance index. This makes the contingency ranking prone to masking problems, where a contingency with many small limit violations is ranked equally well with the one in which there are only a few large limit violations. Also, the selection of weighting factors in the performance index is found to be a difficult task, as it should be chosen based on both the relative importance of buses and branches and the power system operating practice [10]. In addition, majority of the performance indices do not provide an exact differentiation between the secure and non-secure states. The conventionally used performance indices were seen to be calculated separately for line flows and bus voltages, as the overall performance index defined as the sum or weighted sum of the scalar performance indices for bus voltages and line flows could not provide accurate results.

In Ref. [10], authors have proposed an accurate method of critical contingency screening and ranking based on composite security index PI_c which is calculated using Newton Raphson load flow technique. The PI_c is defined based on both bus voltage and line flow limit violations and it has been demonstrated in [10] that it completely eliminates the masking problem. It also provides a proper definition of security in which the secure state is indicated by an index value of '0', while a value greater than '1' indicates an insecure state. Index values lying between '0' and '1' indicate the alarm limit. In this method, the difficult task of selecting the weights is also completely avoided.

In this paper, a cluster computing environment for on-line power system static security assessment based on composite security index PI_c is proposed and a prototype is designed. A variant of the master/slave pattern with only those tools in the open source domain are used for deploying the computational engine. The sequential algorithm for security assessment is parallelized using domain decomposition. Experimental investigations are carried out on IEEE 57 bus demonstrate the effectiveness of the proposed solution.

The outline of the paper is as follows. Formalization of static security assessment problem is given in section II. Development of composite security index is given in section III. A frame work for performing security assessment using cluster computing environment is presented in section IV. Deployment of computational engine along with the definitions of standard performance measures used in parallel/ distributed computing architectures are given in section V. Experimental results and discussions are presented in section VI. Finally conclusions are drawn in section VII followed by references.

II. Power System Security Assessment

Power system security assessment is associated with the steady state and dynamic response of the power system to various disturbances. This process can be divided in to three sequential activities: i. contingency screening and ranking, ii. static and dynamic contingency analysis and iii. preventive and corrective control. The security analysis is performed according to the (n-1) criterion that requires systems to be operated so as to withstand all single contingencies [1]. In this work the first and second activities are mainly considered as they are known to be the bottleneck in the online computations.

A. On line Static Security Assessment

The calculations needed for the on-line static security assessment requires the steady state solution of the power system state equations in order to identify the voltages in all network nodes and the power flows in each line in real time. This real time power flow solution, updated every few minutes, is adopted as reference in the automatic assessment of the static security of the system. The limit violations in bus voltages and line flows identified by computing a scalar performance index each for bus voltages and line flows. Then the solution engine automatically studies hundreds of possible contingencies that would happen on the power system determining how well the system can withstand them [2]. The sequence of major steps for on-line power system static security assessment is as follows:

i. Acquire field data.
ii. A software routine that solves the static power flow problem is invoked. This is then adopted in contingencies analysis as base case study for N configuration.
iii. Check, if the network technical limits are violated. If violated the system is not secure in N configuration.
iv. For each contingency, generate an input file containing the network data modified by the effect of the considered contingency.
v. This file is then used by dedicated software routines to solve the corresponding power flow problem.
vi. Check for each contingency, if the network technical limits are violated.
vii. Generate alarms in the presence of an expected system malfunctioning.

In this work, the violations in network technical limits are identified by computing for base case as well as for each contingency, the composite security index PI_c proposed by the authors in [10], which is defined as a function of both power flow and bus voltage limit violations. Development of composite security index is discussed in the following section.

III. The Composite Security Index

In this paper, the composite security index PI_c developed by the authors in [10] is used for static security assessment. The composite security index has two components one for bus voltage and the other for line flow security check. Two types of limits were defined for bus voltages and line loadings, namely the security limit and the alarm limit. The security limit is the maximum limit specified for the bus voltages and line flows. The alarm limit provides an alarm zone

DOI: 01.IJEPE.4.2.1283

adjacent to the security limit, which gives an indication of closeness to limit violations. The alarm zone also provides a flexible means of specifying the cut-off point for contingency selection based upon numerically ranked security index [10]. It is also possible to treat the constraints on the bus voltage and the line flows as soft constraints, thereby the violation of these constraints, if not excessive, may be tolerated for short periods of time.

The system is considered insecure if one or more bus voltages or line flows exceed their security limit. If one or more bus voltages or line flows exceed their alarm limit without exceeding their security limit, the system is considered to be in the alarm state. If none of the voltages or line flows violates an alarm limit, the system is considered secure. This is indicated by an index value of '0'.

It is assumed that the desirable voltage at each bus *i* is known and is represented as V_i^d. The upper and lower alarm limits and security limits of bus voltages are represented as $F_i^u, F_i^l, V_i^u, V_i^l$ respectively. The normalized upper and lower voltage limit violations of each bus voltage V_i, beyond the alarm limits are defined as in (1).

$$\begin{aligned} d_{v,i}^u &= \frac{V_i - F_i^u}{V_i^d}; \quad if \quad V_i > F_i^u \\ d_{v,i}^u &= 0; \quad if \quad V_i \le F_i^u \\ d_{v,i}^l &= \frac{F_i^l - V_i}{V_i^d}; \quad if \quad V_i < F_i^l \\ d_{v,i}^l &= 0; \quad if \quad V_i \ge F_i^l \end{aligned} \tag{1}$$

For each upper and lower limit of bus voltages, the normalization factor $g_{v,i}$ is defined in (2).

$$\begin{aligned} g_{v,i}^u &= \frac{V_i^u - F_i^u}{V_i^d} \\ g_{v,i}^l &= \frac{F_i^l - V_i^l}{V_i^d} \end{aligned} \tag{2}$$

For power flows, the limit violation vectors d_p and the normalization factor g_p are defined in similar way. Since only the maximum limits are required to be specified for the power flow through each line, two types of upper limits are specified for each line, say the alarm limit P_F and the security limit P_P. The security limit is the specified maximum limit of the power flow through the line. The normalized power flow limit violation vectors for each line *j* can be defined as in (3).

$$\begin{aligned} d_{p,j} &= \frac{|P_j| - P_{F,j}}{Base \quad MVA}; \quad if \quad |P_j| > P_{F,j} \\ d_{p,j} &= 0; \quad if \quad |P_j| \le P_{F,j} \end{aligned} \tag{3}$$

where $|P_j|$ is the absolute value of the power flow through the line *j*.

The normalization factor for each line *j*, is defined in (4) as

$$g_{p,j} = \frac{|P_{P,j}| - P_{F,j}}{Base \quad MVA} \tag{4}$$

For an N-bus, M line system, there are (N+M) dimensional normalized limit violation vectors of both bus voltages and line flows. In multi-dimensional vector space these limit violation vectors form a hyper-box and approximating the hyper-box by a hyper-ellipse inscribed within, a scalar valued index named as composite security index PI_c [10] can be formed. The is defined in (5) as;

$$PI_C = \left[\sum_i \left(\frac{d_{v,i}^u}{g_{v,i}^u} \right)^{2n} + \sum_i \left(\frac{d_{v,i}^l}{g_{v,i}^l} \right)^{2n} + \sum_j \left(\frac{d_{p,j}}{g_{p,j}} \right)^{2n} \right]^{1/2n} \tag{5}$$

where *n* is the exponent used in hyper ellipse equation. From the definition of composite security index, the system is said to be in one of the three states as follows.

- Secure state if the $PI_c = 0$
- Alarm state if $0 < PI_c \le 1$
- Insecure state if $PI_c > 1$

The contingencies can be accordingly ranked in descending order of severity based on PI_c. It is also possible to provide precise information about the buses and/or the lines in which the limit violations occurred so that proper control actions can be taken, without doing a detailed contingency analysis [10].

IV. Power System Static security Assessment Using Cluster Computing Framework

To run the on-line security assessment algorithm described in the previous section, a framework based on cluster computing is proposed in this section, utilizing a hierarchical variant of the master/slave pattern that exploits the hierarchical topology of interconnected computers, such as a network of clusters that can assure high reliability, flexibility, high degree of scalability, and fault tolerance [11].

To reduce the execution time needed for security analysis, a concurrent algorithm based on domain decomposition is adopted, instead of functional decomposition that does not guarantee good performance [12]. In this method the whole job is divided in to similar tasks, each one assigned to a different processor. Since each task coincides with the sequential execution of the analysis of a single contingency, only minor modifications are necessary to the sequential algorithm discussed in section II. The activity diagram in Fig. 1 shows the details of the adopted approach.

The base case and a number of contingencies are analyzed to evaluate the security of the electrical grid. For each case a "slave" task is created, which does the power flow solution for the electrical grid, calculates the composite

DOI: 01.IJEPE.4.2.1283

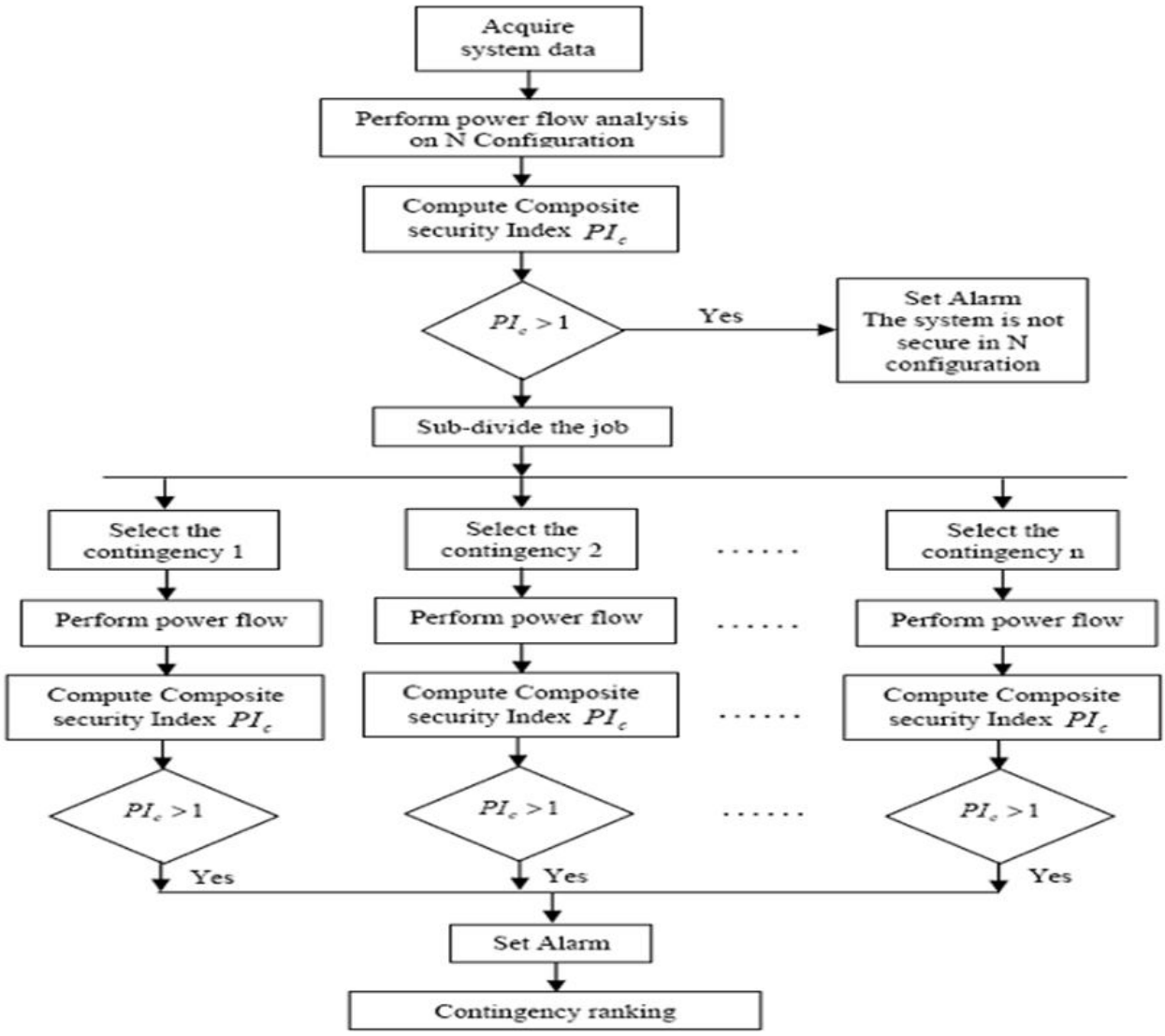

Figure 1. Activity diagram for the Parallelized Security Algorithm

security index PI_c using (5). The critical contingencies are then ranked according to severity based on, and eventually give alarms.

V. Deployment of Computational Engine

The computational engine is deployed on an architecture consisting of Intel® Xeon® CPU's operating at 2.33 GHz, 4096 MB of static RAM. Each of the Xeon CPU is a quad core, implying for every CPU incorporated into the architecture there are 4 effective processors. All the CPU's are connected by a fast Ethernet local area network and are running on CentOS Linux. Only those tools in the open source are used to deploy the computational engine.

A cluster of workstations is formed using the Rocks cluster distribution [13]. The cluster of workstations (COW) can be further extended to cloud computing, as it is also equipped with Eucalyptus to facilitate deploying them as part of into Amazon EC2 based clouds. Ganglia is a scalable distributed system monitor tool for high-performance computing systems such as clusters and grids. It allows the user to remotely view live or historical statistics for all machines that are being monitored.

A. Programming Tools Used

As the principal goal of having the cluster is to parallelize the computation on different machines, protocols must be installed for message-passing for distributed memory applications. In this work, Scilab equipped with Parallel Virtual Machine (PVM) is used for programming on the cluster. Scilab is a numerical computational package developed by researchers from the INRIA and the École nationale des ponts et chaussées (ENPC) [14].

The PVM computing model enables a collection of heterogeneous computer systems to be viewed as a single parallel virtual machine [15]. PVM transparently handles all message routing, data conversion, and task scheduling across a network of incompatible computer architectures. The application is programmed as a collection of cooperating tasks. Tasks access PVM resources through a library of standard interface routines. These routines allow the initiation and termination of tasks across the network as well as

DOI: 01.IJEPE.4.2.1283

communication and synchronization between tasks [16]. The PVM software provides a unified framework in which parallel algorithms can be executed in an efficient and straightforward manner using existing hardware.

B. Performance Measures

Cluster computing involves the execution of a computer program utilizing multiple computer processors concurrently instead of using one processor. In its simplest form, the most obvious benefit of using this architecture is the reduction in execution time of the program. To measure the performance of the proposed computational engine, standard definitions of two types of performance parameters viz. speed up factor, and computational efficiency [12].

The speedup refers to how much a parallel/distributed algorithm is faster than a corresponding sequential algorithm and is defined as

$$S_p = \frac{T_1}{T_p} \tag{6}$$

where p is the number of processors, T_1 is the execution time of the sequential algorithm on one processor and T_p is the execution time of the parallel algorithm with p processors. S_p therefore describes the scalability of the system as the number of processors is increased. Ideal speed up is p when using p processors, i.e. when the computations can be divided in to equal duration processes with each process running on one processor, with no communication overhead.

The efficiency is a performance metric that describes the fraction of the time that is being used by the processors for a given computation. It is defined as

$$E_p = \frac{T_1}{p\,T_p} = \frac{S_p}{p} \tag{7}$$

It is a value, typically between zero and one, estimating how well-utilized the processors are in solving the problem, compared to how much effort is wasted in communication and synchronization.

VI. Experimental Results

The effectiveness of the proposed computational engine for on-line static security assessment is demonstrated through experimental investigations carried out on IEEE 57 bus standard test system and a large IEEE 145 bus 50 machine system. The experiments involved the simulation of credible contingencies such as line outages for each test system for base load condition. Security assessment has been carried out as in [10] by computing the composite security index for the contingencies considered. For programming on the cluster, Scilab equipped with Parallel Virtual Machine (PVM) is used.

For different contingencies, the composite security indices are computed as per (5) and are then ranked in the order of their severity. According to this index, the insecure cases are easily identified as those with values greater than '1' and the secure cases are defined as a value of '0' and can be excluded from the contingency list. If the index value is between '0' and '1', the system is in the alarm state.

Experimental investigations are conducted by varying the number of processors used for computational engine in order to evaluate the reduction in the computation time for the analysis and the performance measures are analyzed. The results obtained for the test system are presented in the following sub-section.

A. IEEE 57 bus test system

The proposed method is applied to IEEE 57-bus test system. The system consists of 57 buses, 72 transmission lines and 8 transformers. The system data and single line diagram for IEEE 57 bus test system has been obtained from [17]. In order to define the composite security index, ±7% and ±10% of the desired bus voltage is taken as the alarm limit and security limit respectively, for each bus. For the line flows, 80% of the thermal limit is chosen as the alarm limit [10].

The contingencies for which a security breach is observed are tabulated in Table I. Column 1 represents different line outage cases. For example**,** L 52-53 represents an outage of line connected between bus numbers 52 and 53. In Column 2, the composite security index PI_c computed for the corresponding contingency case is presented and the security status is shown in column 3. In this case line outages L 52-53 and L 14-15 are found insecure for which the value is greater than '1'.

Table I. Contingency Ranking for IEEE 57 Bus Test System

Contingency (Line Outage)	**Composite Security Index**	**Security Status**
(1)	**(2)**	**(3)**
L 52-53	8.714	Insecure
L 14-15	3.925	Insecure
L 25-30	0.9128	Alarm state
L 1-17	0.5487	Alarm state
L 5-6	0. 0531	Alarm state
L 1-2	0	Secure
L 2-3	0	Secure
L 4-6	0	Secure
L 6-8	0	Secure

For the line outages L 25-30, L 1-17, and L 5-6 the system is found to be in the alarm state with indices between '0' and '1'. For all other contingencies the system is found secure with an index value of '0'. The remaining contingency cases which are actually secure are not shown in the Table.

The total execution time or turnaround time taken for a single processor architecture and multi-processor architecture with number of processors (*P*) in increments of two is tabulated in columns 2 of Table II, for both programming tools. The computational times are arrived at by performing the computation several times and the average has been taken. Percentage reduction in execution time is also analyzed and tabulated in column 3.

DOI: 01.IJEPE.4.2. 1283

TABLE II. EXECUTION TIME TAKEN FOR IEEE 57 BUS SYSTEM

P	Execution Time in seconds	% reduction in execution time
(1)	(2)	(3)
1	44.639	0
2	24.392	45.4
4	15.498	65.3
6	12.682	71.6
8	8.982	79.9
10	7.096	84.1
11	6.404	85.7

It has been shown that, for IEEE 57-bus test system, the turnaround time required for complete static security assessment with a simulation engine based on single processor architecture is 44.639 seconds. If multi-processor architecture is used for computation, the turnaround time gets considerably reduced. For example with 6 numbers of processors the total execution time obtained is 12.682 seconds with a reduction in computation time of 71.6 % with respect to the single processor architecture.

B. IEEE 145 bus 50 Machine system

In order show the effectiveness of the proposed computational engine to reduce the computational overhead of large power systems, cluster computing approach was also applied to IEEE 145 bus 50 machine test system [17] for static security assessment. The system consists of 453 transmission lines including 52 fixed tap transformers.

Static security assessment has been carried out by computing the composite security index for post contingency conditions taking line outages as contingencies. For each bus, ±7% and ±10% of the desired bus voltage is taken as the alarm limit and security limit respectively. For the line flows, 80% of the thermal limit is chosen as the alarm limit. Contingency cases are ranked in the decreasing order of severity based on the composite security index and are shown in Table III. It can be observed that around 43 contingencies are identified as insecure with index values greater than '1'. The contingency cases that are actually in secure state or in alarm state are not shown in the Table.

The total computational time for a single processor and by varying the number of processors in increments of two is tabulated in Table IV. Explanations can be made similar to that of IEEE 57 bus system.

From Table IV it can be observed that, for large IEEE 145 bus system, the complete static security assessment took 1343.2 seconds with single processor architecture. The execution time gets considerably decreased with the multi-processor architecture as shown in Table IV. Percentage reduction in computation time is analyzed and is provided in column 3 of Table IV. With an 11 processor architecture it can be observed that the computation time get decreased by 87.8% with respect to single processor architecture. So the proposed computational engine is scalable.

TABLE III. CONTINGENCY RANKING FOR IEEE 145 BUS TEST SYSTEM

Sl. No	Outage	Composite security index PI_C	Security Status
1	L 119-128	5811	
2	L 121-125	5807	
3	L 121-127	5770	
4	L 125-129	5770	
5	L 102-117	3745	
6	L 134-145	511.0576	
7	L 141-145	227.1176	
8	L 108-121	182.1201	
9	L 139-141	157.4557	
10	L 134-144	130.9143	
11	L 136-145	118.8362	
12	L 17-59	110.4603	
13	L 72-112	99.1629	
14	L 27-75	93.089	
15	L 67-124	80.0822	
16	L 65-66	78.857	
17	L 141-115	58.9581	
18	L 6-7	38.4065	
19	L 142-131	32.8291	
20	L 24-76	31.3636	
21	L 136-115	27.5093	
22	L 131-132	21.7238	Insecure
23	L 120-125	19.8099	
24	L 36-99	15.0637	
25	L 122-125	14.9502	
26	L 25-27	13.1515	
27	L 142-144	12.6587	
28	L 136-142	10.7621	
29	L 137-145	10.1162	
30	L 142-143	8.1974	
31	L 66-69	8.0106	
32	L 73-105	6.5935	
33	L 74-106	6.0114	
34	L 119-120	5.743	
35	L 66-111	3.0649	
36	L 1-6	2.301	
37	L 2-6	2.2583	
38	L 7-104	2.0933	
39	L 33-110	1.764	
40	L 1-2	1.7066	
41	L 120-128	1.4103	
42	L 6-12	1.2981	
43	L 144-145	1.1321	

TABLE IV. EXECUTION TIME TAKEN FOR IEEE 145 BUS SYSTEM

P	Execution Time in seconds	% reduction in execution time
(1)	(2)	(3)
1	1343.18	0
2	746.21	44.4
4	408.6	69.6
6	317.82	76.3
8	232.46	82.7
10	181.43	86.5
11	163.41	87.8

DOI: 01.IJEPE.4.2.1283

A. Performance Comparison

The performance of the computational engine for IEEE 57 bus and IEEE 145 bus test systems are analyzed by computing the performance parameters viz. speed up and computational efficiency as per (6) and (7) respectively. The variation of speedup and computational efficiency with the number of processors, are given in Fig. 2 and Fig. 3 respectively for both tests systems.

The speedup factor S_p is increasing consistently with the increase in the number of processors for both test systems. This implies that the performance improvement is guaranteed and hence the proposed system is scalable. It can be observed from Fig. 3 that, as the number processors increased above 6 the computational efficiency also get improved.

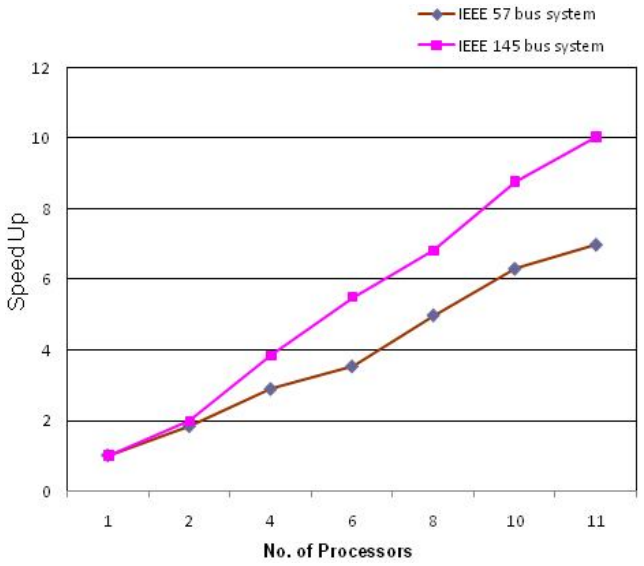

Figure 2. Speed up curve for IEEE 576 bus and IEEE 145 bus systems

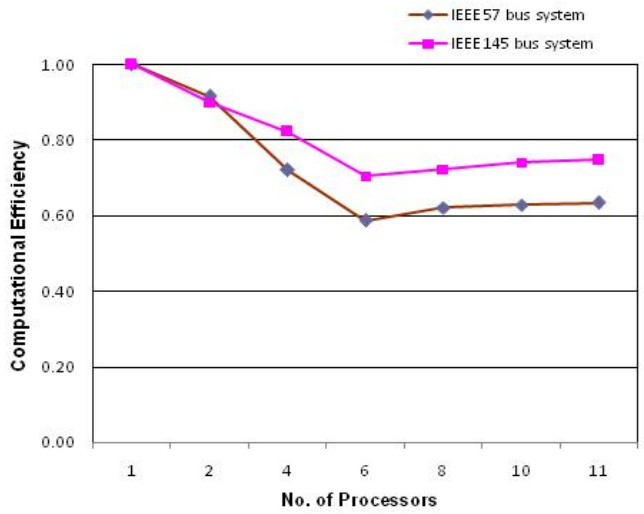

Figure 3. Computational Efficiency Curve for for IEEE 576 bus and IEEE 145 bus systems

It can also be observed that as the computational complexity increases as in the case of large IEEE 145 bus system, speed up and computational efficiency are also increased. This ensures the scalability of the system and the maximum utilization of processor's capacity in solving the problem compared to the time wasted for communication and synchronization. It can be concluded that depending upon the size and complexity of the power system, more number of processors can added, so that the security status can be obtained within the stipulated time, and proper control actions can be suggested to bring the system back to the secure state.

VII. Conclusion

A scalable solution based on high performance computing clusters is presented to address the requirement for faster and accurate methodologies for real time power system security analysis. The developed composite security index based on both power flow and bus voltage limit violations acts as the tool on-line static security assessment. Investigations were carried out different standard test systems, to validate the proposed method. The preliminary investigations reveals that high performance computing engine based on COW's is a viable scalable solution for performing accurate and fast security analysis. The efficiency of the engine increases with the complexity of the system. This is a first step towards realizing the full potential of cluster computing architecture for power system static security assessment. The COW, being equipped with Eucalyptus, can be integrated into private or public clouds making use of the computing resources. As the cloud computing has the potential to furnish on-demand a dynamically variable computational power without affecting the accuracy, the research in this direction is under progress. The work can also extended for dynamic security assessment for predicting the impact of critical contingencies on the system.

References

[1] K. Morison, "Power system security in the new market environment: future directions," Proc. IEEE-PES Winter Meeting, pp. 78–83, 2000.

[2] Quirino Morante, Alfredo Vaccaro, Domenico Villacci and Eugenio Zimeo, "A web based computational architecture for power systems analysis", Proc. of the International Conference on Bulk Power System Dynamics and Control- VI, Cortina d'Ampezzo, Italy, Aug.22-27, pp.240-246, 2004.

[3] Neal J Balu et. al., "On-line power system security analysis," Proc. of the IEEE, Vol.80, no.2, pp. 262-280, Feb. 1992.

[4] Stott B, Alsac O and Monticelli A J, "Security analysis and optimization", Proc. of the IEEE, Vol. 75, no.12, pp.1623-1644, Dec. 1987.

[5] T S Sidhu and L Cui, "Contingency screening for steady-state security analysis by using FFT and artificial neural networks," IEEE Transactions on Power Systems, Vol.15, pp. 421–426, Feb. 2000.

[6] V Brandwajn, A B R Kumar, A Ipakchi, A Bose, and S D Kuo, "Severity indices for contingency screening in dynamic security assessment," IEEE Transactions on Power Systems, Vol. 12, pp. 1136–1142, Aug. 1997.

[7] Daniel J Tylavsky and Anjan Bose, "Parallel processing in power systems computation," IEEE transactions on Power Systems, Vol. 7, no. 2, pp. 629-638, May 1992.

DOI: 01.IJEPE.4.2.1283

[8] O R Saavedra, "Solving the security constrained optimal power flow problem in a distributed computing environment," IEEE Proceedings on Generation Transmission and Distribution, Vol. 143, no. 6, pp. 593–598, Nov. 1996.

[9] A B Alves, A Monticelli, "Static security analysis using pipeline decomposition," IEEE Proceedings on Generation Transmission and Distribution, Vol. 145, no.2, pp.105–110, Mar. 1998.

[10] Sunitha R, Sreerama Kumar R., Abraham T. Mathew, "A Composite Security Index for On-line Static Security Evaluation". Int. national Journal of Electric Power Components and Systems, Taylor & Francis, Vol.39, no.1, pp 1-14, Jan. 2011.

[11] V C Ramesh, "On distributed computing for on-line power system applications" Electrical power and energy systems Elsevier science limited, Vol. 18, no. 8, pp.527-533, Mar. 1996.

[12] G Aloisio, M l Scala, and R Sbrizzai, "A distributed computing approach for real time transient stability analysis," IEEE transactions on Power Systems, Vol.12, pp. 981–987, May 1996.

[13] University of California, "Rocks Base Roll: Users Guide," version 5.2 edition August 2009.

[14] Introduction to Scilab. User's guide [Online]. Available: http://cermics.enpc.fr/scilab_new/site/Liens/intro/intro.html

[15] Al Geist et.al. PVM: Parallel Virtual Machine A Users- Guide and Tutorial for Networked Parallel Computing, The MIT Press, Cambridge, 1994. [Online]. Available:http://www.csa.ru

[16] Michael J Quinn, Parallel Programming: Using MPI and OpenMP – Tata McGraw -Hill Edition 2003

[17] Power System Test case Archive. [Online]. Available: http://www.ee.washington.edu/research/pstca/

DOI: 01.IJEPE.4.2.1283